What is claimed is:

1. A method for dynamically partitioning a storage system cache among multiple workload classes having different quality-of-service (QoS) requirements, the cache holding data as data pages, the method comprising the steps of:

maintaining a history of recently evicted pages for each class;

determining a future cache size for the class based on the history and the QoS requirements, the future cache size being different than a current cache size for the class;

determining whether the QoS requirements for the class are being met; and

adjusting the future cache size to maximize the number of classes in which the QoS requirements are met.

2. The method as recited in claim 1, wherein the step of determining whether the QoS requirements for the class are being met includes the steps of:

recording data concerning a QoS requirement for the class; and

comparing the recorded data with said QoS requirement.

3. The method as recited in claim 1, wherein the step of determining a future cache size includes the steps of:

recording cache hit data in the history of the class;

recording the cache size corresponding to the cache hit data; and

determining the future cache size based on the cache hit data and the respective cache sizes.

4. The method as recited in claim 1, wherein the step of adjusting the future cache size includes the steps of:

increasing the future cache sizes of the classes whose QoS requirements are not met; and

decreasing the future cache sizes of the classes whose QoS requirements are met.

5. The method as recited in claim 4, wherein the step of increasing the future cache sizes includes the step of setting a future cache size as a function of current cache size and the difference between cache hit data and corresponding cache sizes.

6. The method as recited in claim 4, wherein the step of decreasing the future cache sizes includes the step of setting a future cache size as a function of the current cache size, the number of classes and the difference between cache hit data and corresponding cache sizes.

7. The method as in claim 1 further comprising the step of allocating the cache space to the classes to maximize the overall cache hits if the QoS requirements for all classes are met.

8. The method as recited in claim 1, wherein the future cache size is adjusted periodically.

9. The method as recited in claim 1, wherein the future cache size is adjusted continuously on very request for data.

10. The method as recited in claim 1, wherein the the future cache size is adjusted to maximized the total class objectives.

11. A storage system capable of dynamically partitioning a system cache among multiple workload classes having different quality-of-service (QoS) requirements, the cache holding data as data pages, the system comprising:

means for maintaining a history of recently evicted pages for each class;

means for determining a future cache size for the class based on the history and the QoS requirements, the future cache size being different than a current cache size for the class;

means for determining whether the QoS requirements for the class are being met; and

means for adjusting the future cache size to maximize the number of classes in which the QoS requirements are met.

12. The system as recited in claim 11, wherein the means for determining whether the QoS requirements for the class are being met includes:

means for recording data concerning a QoS requirement for the class; and

means for comparing the recorded data with said QoS requirement.

13. The system as recited in claim 11, wherein the means for determining a future cache size includes:

means for recording cache hit data in the history of the class;

means for recording the cache size corresponding to the cache hit data; and

means for determining the future cache size based on the cache hit data and the respective cache sizes.

14. The system as recited in claim 11, wherein the means for adjusting the future cache size includes:

means for increasing the future cache sizes of the classes whose QoS requirements are not met; and

means for decreasing the future cache sizes of the classes whose QoS requirements are met.

15. The system as recited in claim 14, wherein the means for increasing the future cache sizes includes means for setting a future cache size as a function of the current cache size and the difference between cache hit data and corresponding cache sizes.

16. The system as recited in claim 14, wherein the means for decreasing the future cache sizes includes means for setting a future cache size as a function of the current cache size, the number of classes, and the difference between cache hit data and corresponding cache sizes.

17. The system as recited in claim 11 further comprising means for allocating the cache space to the classes to maximize the overall cache hits if the QoS requirements for all classes are met.

18. The system as recited in claim 11, wherein the future cache size is adjusted periodically.

19. The system as recited in claim 11, wherein the future cache size is adjusted continuously on very request for data.

20. The system as recited in claim 11, wherein the the future cache size is adjusted to maximized the total class objectives.

21. A computer-program product for use with a storage system for dynamically partitioning a system cache among multiple workload classes having different quality-of-service (QoS) requirements, the cache holding data as data pages, the computer-program product comprising:

a computer-readable medium;

means, provided on the computer-readable medium, for maintaining a history of recently evicted pages for each class;

means, provided on the computer-readable medium, for determining a future cache size for the class based on the history and the QoS requirements, the future cache size being different than a current cache size for the class;

means, provided on the computer-readable medium, for determining whether the QoS requirements for the class are being met; and

means, provided on the computer-readable medium, for adjusting the future cache size to maximize the number of classes in which the QoS requirements are met.

22. The computer-program product as recited in claim 21, wherein the means for determining whether the QoS requirements for the class are being met includes:

means, provided on the computer-readable medium, for recording data concerning a QoS requirement for the class; and

means, provided on the computer-readable medium, for comparing the recorded data with said QoS requirement.

23. The computer-program product as recited in claim 21, wherein the means for determining a future cache size includes:

means, provided on the computer-readable medium, for recording cache hit data in the history of the class;

means, provided on the computer-readable medium, for recording the cache size corresponding to the cache hit data; and

means, provided on the computer-readable medium, for determining the future cache size based on the cache hit data and the respective cache sizes.

24. The computer-program product as recited in claim 21, wherein the means for adjusting the future cache size includes:

means, provided on the computer-readable medium, for increasing the future cache sizes of the classes whose QoS requirements are not met; and

means, provided on the computer-readable medium, for decreasing the future cache sizes of the classes whose QoS requirements are met.

25. The computer-program product as recited in claim 24, wherein the means for increasing the future cache sizes includes means, provided on the computer-readable medium, for setting a future cache size as a function of the current cache size and the difference between cache hit data and corresponding cache sizes.

26. The computer-program product as recited in claim 24, wherein the means for decreasing the future cache sizes includes means, provided on the computer-readable medium, for setting a future cache size as a function of the current cache size, the number of classes, and the difference between cache hit data and corresponding cache sizes.

27. The computer-program product as recited in claim 21 further comprising means, provided on the computer-readable medium, for allocating the cache space to the classes to maximize the overall cache hits if the QoS requirements for all classes are met.

28. The computer-program product as recited in claim 21, wherein the future cache size is adjusted periodically.

29. The computer-program product as recited in claim 21, wherein the future cache size is adjusted continuously on very request for data.

30. The computer-program product as recited in claim 18, wherein the the future cache size is adjusted to maximized the total class objectives.